Detection of suspected nodes in MANET

Suman S Chandanan[1], Brajesh Patel[2], Amit Kumar Chandanan[3]
[1]Shri Ram Institute of Technology, Jabalpur, India
[1]Email: suman_singh2008@yahoo.com
[2]Shri Ram Institute of Technology, Jabalpur, India
[2]Email: brajesh.patel@rediffmail.com
[3]Hitkarini College of Engineering and Technology, Jabalpur, India
[3]Email: amit.chandanan@hcet.hitkarini.com

***Abstract*—Mobile ad hoc network (MANETs) is an emerging area with practical applications. One such field concerns mobile ad hoc networks (MANETs) in which mobile nodes organize themselves in a network without the help of any predefined infrastructure. Securing MANETs is an important part of deploying and utilizing them, since them are often used in critical applications where data and communications integrity in important. Existing solutions for wireless networks can be used to obtain a certain level of such security. Nevertheless, these solutions may not always be sufficient, as ad-hoc networks have their own vulnerabilities that cannot be addressed by these solutions. To obtain an acceptable level of security in such a context, traditional security solutions should be coupled with an intrusion detection mechanism. We propose using a quantitative method to detect intrusion in MANETS with mobile nodes. Our method is a behavioral anomaly based system, which makes it dynamic, scalable, configurable and robust. Finally, we verify our method by running ns2 simulations with mobile nodes using Ad-hoc on-demand Distance Vector (AODV) routing. It is observed that the malicious node detection rate is very good, and the false positive detection rate is low.**

***Keywords*- MANET, Intrusion detection, AODV**

I. Introduction

Misbehaving nodes in a MANET can adversely the availability of services in the network shown by the research [3].node misbehave either because they are broken, selfish or malicious. Broken nodes are non functional node in the network. A node is agreed to forward traffic on the behalf of other node but it work as non functional node prior to it fulfill this agreement. Selfish node can agree to forward packet but silently drop the packets in attempt to consume bandwidth and energy of the channel. the decentralized nature, scalable setup and dynamically changing topology makes ad hoc networks ideal for a variety of applications ranging from front-line zones(military and natural) to data collection as investigated in [4],[8], [16].Number of MANET secure routing schemes in the research literature, for the example [5], [6], [7], [10] do not mitigate against these misbehaviors. In this paper we present quantitative intrusion detection mechanism on-demand multipath source routing protocol that effectively mitigate against selective packet dropping effect and other adversarial activities.

The main contribution of this paper is concept of intrusion detection mechanism used to discourage selfish and adversarial behavior in MANET.

II. Problem statement

Intrusion Detection is an activity that determines whether a process or user is attempting something unexpected. It works, as defined by [10], on the basis of examining activity on a specific machine or network and deciding whether the activity is normal or suspicious. It can either compare current activity to known attack patterns or simply raise an alarm condition when specific measurements exceed preset values.There have been many approaches intrusion detection in MANETs.

The initial classification is based on authentication based schemes. These rely on the identification of nodes by a unique identifier. Use of encryption keys fall into this category, and they have been deeply studied. The second approach is behavioral based algorithms where intrusion is defined based on nodal activities, rather than its identifier. This, according to us, is a better approach for the following reasons:

- Node identities can be easily stolen. Behavior is tougher to replicate.
- Identity based behavior involves storage of Identifier databases or logic
- Each new node has to be given a unique identifier, making the process of deployment more expensive (time and cost).

Thus, we limit our focus to intrusion detection based on behavior, since we think it is a more efficient, lightweight and easily scalable solution to Intrusion Detection in MANETs. Intrusion Detection Systems based on behavior can be broadly classified into these categories: anomaly detection, signature or misuse detection, and specification based detection. We mention these as per the taxonomy proposed in [11].

A. Anomaly Detection

In such systems, a baseline profile of normal system activity is created. Any system activity that deviates from the baseline is treated as a possible intrusion. The problems with this approach are:

- Anomalous activities that are not intrusive are flagged as intrusive(false positives)

DOI: 01.IJNS.03.01. 61

- Intrusive activities that behave in a non-anomalous manner are not detected (false negatives)

Anomaly detection for mobile computing may demand that the normal profile be periodically updated and the deviations from the normal profile computed. The periodic calculations can impose a heavy load on some resource constrained mobile devices.

Zhang and Lee [15] propose distributed and cooperative intrusion detection model in their pioneer work in this field based on statistical anomaly detection techniques. Every node in the network participates, and runs an IDS agent runs which performs local data collection and local detection, whereas cooperative detection and global intrusion response can be triggered when a node reports an anomaly. The authors consider two attack scenarios separately - abnormal updates to routing tables, and detecting abnormal activities in layers other than the routing layer; these formed the definition of the anomaly.

B. Misuse Detection

In misuse detection, decisions are made on the basis of the signature of an intrusive process, and the traces it leaves in the observed system. Legal behavior is defined and observed behavior compared against it to recognize intrusions. Such a system tries to detect evidence of intrusive activity irrespective of any knowledge regarding the background traffic (i.e., the historical behavior of the system). In an architecture proposed by P. Albers, O. Camp etc, [18], the authors suggest using nodes individually running misuse-detecting local IDS (or LIDS) agents. They define misuse/ attack signatures using variables in SNMP Management Information Bases (MIB) variables. A prototype that defines misuse as telnet access arising from outside the community has been tested.

C. Specification Based Detection

This defines a set of constraints that describe the correct operation of a program or protocol, and monitors the execution of the program with respect to the defined constraints. This technique may provide the capability to detect previously unknown attacks, while exhibiting a low false positive rate.

Tseng, Balasubramanyam et al [13] propose IDS based on this approach. Their approach uses finite state machines to specify correct AODV routing behavior and distributed network monitors for detecting run-time violation of the specifications .Similar work for DSR has been done by P. Yi, Y .Jiang at al [14].

D. Compound Detection

An improvement over misuse and anomaly detection is compound detection, which is misuse inspired system that forms a compound decision based on both the normal behavior of the system and the intrusive behavior of the intruder. The detector operates by detecting the intrusion against the historical, normal traffic in the system. These detectors are said to have a greater accuracy in detecting undefined behavior. They would at the very least be able to qualify their decisions better. M. Alam, T Li et al, in [12], proposes an IDS which uses a quantitative method of anomaly definition based on transmission characteristics, but factors in historical transmission behavior of the node.

III. Problem difination

Joo B. D. Cabrera, Raman K. Mehra [19] defines an "ensemble" method to detect, report and average anomaly-data in networks using clusters. Each node runs a"Local IDS", and measures Anomaly-index measuring the deviation of measured data from the normal. This is reported to the cluster head which then propagates a cluster level anomaly index to a manager, which performs all the decisions.

It is a distributed Solution, but involves two levels of central entities. The presence of central entities makes it a central point of failure - the cluster may become dysfunctional if an attacker targets the cluster head. Also, these central nodes are usually more resource intensive (also due to complex logic), and decrease survivability.

M. Alam, T Li et al in [12] suggests a non-centralized solution, but do not cater to mobile nodes or MANETs. Our central challenge is to find a quantitative, distributed and dynamic intrusive detection solution for MANETs that involve mobile nodes in a non-cluster based environment.

A. Specific Needs and Challenges

This section breaks up our research problem definition into further detail. This will assist in proposing a solution that addresses each of the challenges or problems faced in creating an intrusion detection system.

B. Intrusion Detection

Members of MANETS that display erroneous or malevolent behavior are often termed "malicious" nodes; from now on, all nodes that display any undefined or unexpected behavior are referred to as "malicious nodes". Thus, the first question to be answered is: How do we identify nodes displaying malicious behavior? In other words, how is the anomaly described?

Secondly, nodes moving in uncontrolled environments with relatively poor physical protection have a non-negligible probability of being compromised. Along with attacks from the outside world, the possibility of attacks launched by compromised nodes from within the network exists too. So, the second question to answer is: Is our solution time-continuous? Can a node that started as a legal node, but was compromised after some time is recognized?

IV. Research approach

The solution to research challenge is presented in this section. It is based on the quantitative intrusion detection techniques in [24], but is applied to a MANET containing mobile nodes.

DOI: 01.IJNS.03.01. 61

A. Research Approach Features

The detailed set of points against which to measure the effectiveness of proposed approach were mentioned in section 3.1.1, The answers to those questions are enumerated here.

- Is our solution time-continuous? Can a node that started as a legal node, but was compromised by another malicious node be recognized? Yes, our solution bases only on transmission and response behavior alone. Our solution is also dynamic; every node regularly (periodically) checks its neighbor statistics to determine abnormal behavior.
- Is our proposed method truly distributed? Yes. Each node in the MANET, or any number of nodes in the MANET can be configured to assume the responsibility of detecting abnormal behavior

B. Detection of Suspected Node

The first level of moving toward a secure ad hoc network consists of identification of nodes within the network that display unexpected behavior, or, in other words, may have turned malicious. Identifying malicious nodes consists of two steps.

The first is the recognition of nodes that may be classified as displaying malicious behavior, and the second is to ascertain whether that classification is correct.

- Recognition: Detecting malicious nodes entails defining the term malicious. The scope of the current research allows definition of malicious nodes as those that have aberrations in data exchange patterns. Dr. Alam, Tao Li et al, in [12], propose a method in which nodes are expected to acknowledge every message it receives. Every node measures the number of acknowledgments it has received from the neighbor nodes 1 it has tried to transmit to.

In other words, each node records the throughput of every neighbor node it has attempted to communicate with. This value is a measure of near-term behavior. This behavior measured over a period of time determines the historical quality of behavior of the neighbor node. This statistic is the stability of the nodal behavior, and will henceforth be referred to as"STB ()". "Data transmission quality" (referred to as DTQ from now on) is defined as a function of STB (), probability of error in the channel (P ()), and the energy needed to transmit data (E).

$$DTQ = k \times \frac{D \times STB()}{E \times P()} \quad (1)$$

D = Power needed for transmitting the total data attempted to be sent,
E = Energy needed to send 1 byte of data, and
k is a constant.

The current research is limited to non-cluster based networks, and it varies here from the paper mentioned above. Also, in the current research, transmission is always atomic in terms of packets - a packet is either transmitted completely, or not at all. We rely on measuring the effectiveness of transmission from a node to another. Each node calculates and maintains DTQ for each of its neighbor nodes.
When DTQ value falls below set thresholds, the neighbor is signaled as a malicious node.

- Confirmation: The next step in the identification of such nodes is to ascertain whether a reading made by one node is correct. This is decided based on a group consensus approach every node in the network is sent a request to accept/reject this decision.

Nodes receiving such a request can vote for or veto by referring to its own DTQ readings for the node in question. The vote initiating node then draws a consensus based on these replies. If more votes have been received approving of malicious behavior, the node is added into a Black-list that allows all nodes to refrain from further communication with this node.

C. Recognition of Malicious Nodes

Recognition of a node displaying malicious behavior is a continuous process followed by each node. The process of malicious node recognition is detailed by the flowchart in fig.1.

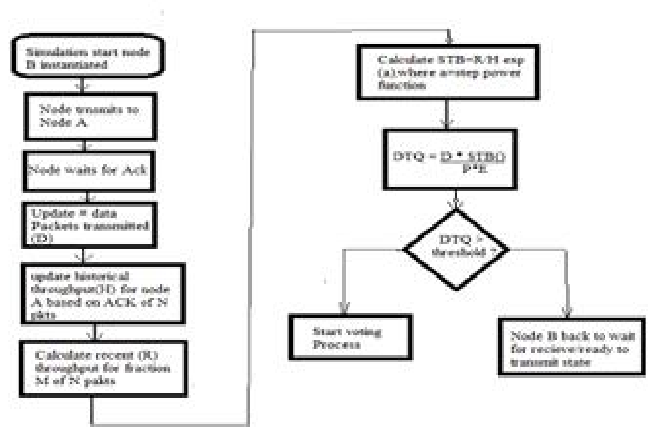

Figure 1. Flow chart for Intrusion detection

D. Confirmation of Detection

The next step is that of collectively deciding whether a node whose behavior is erratic is actually a malicious one. For example, node A has detected that node B's DTQ has fallen below a threshold.

Node A now wants consensus on its suspicion, and triggers a vote by sending a broadcast request for the same. When MANET nodes receive such a request, they check the DTQ values for node B in their tables, and reply with a positive or negative vote. These votes are aggregated at node A to decide node B's status.

E. Voting Details

Its flow is defined through flowchart in fig.2. A few more details within the voting process are discussed below.

DOI: 01.IJNS.03.01. 61

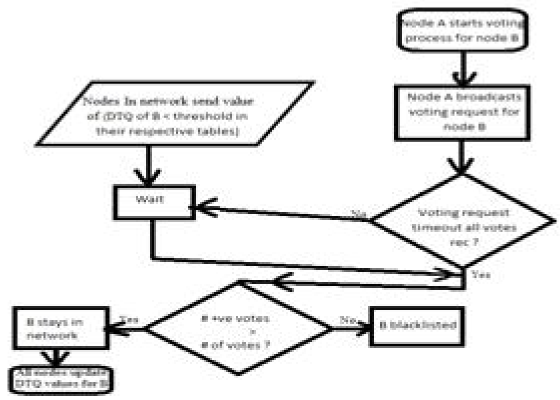

Figure 2. flow chart for voting process

(a) Vote arrival: A vote-initiating node keeps a count of the number of votes in receives. It also does not register more than 1 vote from the same neighbor, for a particular vote-request. Once it has received votes from all of its neighbors, it decides for or against the voted-upon node. For this implementation, we take "all" neighbors to mean the total number of node in the network less one, which is the maximum expected neighbor count.

(b) Vote Request timeout: The situation where all neighbors respond is an ideal situation, in wireless networks and more so in MANETs, where data packets may be lost in transit. Also, some nodes may decide not to vote. In such cases, the vote-initiator cannot wait indeterminately. The vote request time out solves this dilemma, and is set as soon as the vote-request is sent out. At the end of this time-out period, the vote request initiator aggregates all the votes it has received, and makes a decision based on the counts. All votes received after this timeout are useless.

(c) Who votes? All nodes that receive a vote-request attempt to vote. However, if the number of messages they receive from the vote-initiator is not sufficient for them to decide, they refrain from voting. We will discuss this sufficiency number in sections that follow.

(d) Process after vote decision:

i. On blacklisting: Immediately after a node has been blacklisted, as demonstrated in fig. 4, a message is sent out to all nodes with this information. All nodes receiving this message add the node to their blacklist details too. Once a node is blacklisted, no communication from such nodes is responded to anymore.

ii. On being acquitted: If a node is acquitted after the vote decision, all nodes treat it as usual. No information about the acquittal is sent out. This raises the question as to whether the vote-initiator, who now has a low DTQ values for this node will repeatedly generate redundant vote-requests! In short, the answer is no! The vote request is scheduled only once every bucket.

So, if it fails in one bucket, then, the node waits for the next bucket to occur before it can make a new vote request. Within this bucket, if communication with the node in question improves, then no vote-requests are rescheduled. If not, the node may genuinely be a malicious one, and it's time to ask for a consensus. This guarantees that there are no premature, repeated vote requests.

E. Acknowledgment for Messages

Acknowledgment messages every node sends an acknowledgment of message receipt as soon as a it receives a data message. The sender waits for acknowledgment for some time.

(a) Acknowledgment arrival: If the acknowledgment arrives on time, the statistics for the acknowledgment sender are updated. If this is the end of a bucket, the DTQ is calculated anew, and a comparison for DTQ versus threshold is made. If necessary, a vote-request is scheduled.

(b) Acknowledgment timeout: The ACK-timeout is the time a sender A waits for an acknowledgment from the intended recipient, node B .If the acknowledgment does not arrive on time (i.e. arrives after ACK-timeout seconds), and if this is the end of a block, then, again, the DTQ is recalculated and the process of comparison repeats. Also, if the end of a block (bucket) is reached, the sender no longer accepts any more acknowledgments for this block of sent data, i.e. the DTQ for this block of data is final.

V. Simulation and result analysis

This section is dedicated to represent graphical simulated result and their analysis in NS-2[1]. The simulation is aim to show the performance of the routing protocols with present of Detected malicious nodes in the mobile ad hoc network for this two metrics are simulated : first, change in the mobility futures of the nodes and second, changes in the IDS settings of the nodes.

A. Changes in Mobility Features

This section aims to measure the functioning of our IDS when changing features of mobility like speed.

A.a . Simulation Results

Here, in this section all simulated results shown with their parameters

A.a.a Changes in the Mobility features

This set of tests measure the functionality by varying speeds of all nodes uniformly, and then heterogeneously.

A.a.a.a. Varying Speed

Table I. Simulation Parameters for mobility

Name	Value
Number of Nodes	10
Simulation run time	400 seconds
Mobility update Interval	1 seconds
Malicious node count	4
Malicious node ID	Node 4,5,6,8
ACK timeout	30 seconds
Initial speed	5 m/s
History count bucket	20
Number of buckets	2

DOI: 01.IJNS.03.01.61

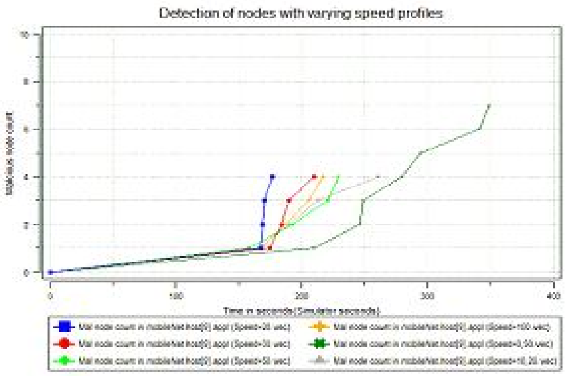

Figure 3. Intrusion Detection with heterogeneous speeds

Fig.3 displays the count of nodes that are recognized as a function of time.

A.A.B Changes in IDS settings

We run the various test below with the below set of common configuration details. The sections then indicate the changed details alone.

A.A.B.A Varying History count and bucket count

Table II. Settings used for IDS configuration tests

Name	Values
Mobility update interval	1 seconds
Malicious node count	5
Malicious node ID	0,1,5,6,12
ACK timeout	30 seconds
Initial speed of nodes	5 m/s
History Count bucket	20
Number of buckets	2

The following fig.4 shows the effect of change in near-time buckets

A.A.C Changes in network setup

This section seeks to measure the behavior of our IDS when the topology of the network it is being used changes. This may pertain to the number of malicious nodes introduced, the number of nodes employed by the network as a whole etc.

A.A.C.A Number of malicious nodes in the network

This section aims to check the effectiveness of our IDS in tracing malicious nodes independent of the number of such nodes present in the system. The tests are conducted by using a varying count of malicious nodes, perpetrating 20 to 90 percent of the network (20, 40, 60, 80 and 90 percent) fig. 5 captures the result.

A.A.C.B Number of nodes in the network

We have shown that the correct number of malicious nodes, and the exact malicious nodes are pointed out, whatever be the configuration of the number of nodes in the network. Thus, we are not repeating tests for this section.

B. Discussion

All malicious nodes are successfully detected.

- There is possibility of false positives, as noticed.
- False positives can be explained by analyzing the sent/ received/ acknowledged message counts for various nodes using the output files .False positives occur due to one of the following reasons:

A node, say, A, does not receive messages for an extended period from a particular node, say B. The sending node B evaluates the absence of acknowledgments from A as malicious behavior, even though A is a legal node (i.e. based on our simulation settings). However, this is good behavior, since we have positively identified nodes based on their transmission characteristics, and can identify innocent nodes that have turned malicious after establishment of the network. Vote-replies are lost. Why or where are messages lost?

Either in transit, or due to time-out due to losing connectivity while being mobile. The former happens because the routing, Mac and physical layers use (the AODV [17] based implementation) have a definite loss factor, which increases with speed. This is documented by the AODV implementation documentation - tests for the AODV implementation were done for speeds not exceeding of 10 mps, and with not more than 25 nodes.

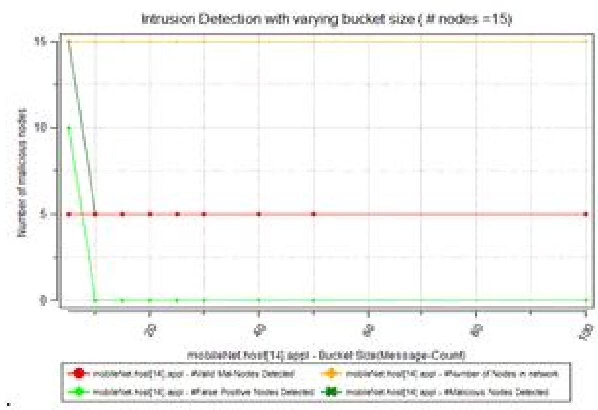

Figure 4. Detection rate with varying near-term bucket size

All malicious nodes are detected correctly. It is noticed that when the near-time bucket size is low, the false positive detection rate is high. This is expected, because a low near-time-bucket value means that the behavior of the nodes is measured based on very few transmissions (and the acknowledgments received for fewer transmissions). As the bucket size becomes more in tune to the network's current settings of behavior, false positives become almost nil.

The second effect is that of having a low history count itself. This also displays the same behavior as that above. This is because long-term-bucket measurements aim to capture the long term behavior of nodes. Say, historically, a node has an 80 percent acknowledgment rate.Then, using near-term buckets, we measure if the node is consistent with its"character" of 80 percent. If not, the activity is of interest and may be marked for a vote-request trigger. But, if the period over which history is measured is lowered (by reducing the number of transmissions we monitor), it does not present a true measure of regular node behavior.The graph starts with 20 percent perpetration and proceeds to 90 percent perpetration. All malicious nodes are successfully detected.

DOI: 01.IJNS.03.01. 61

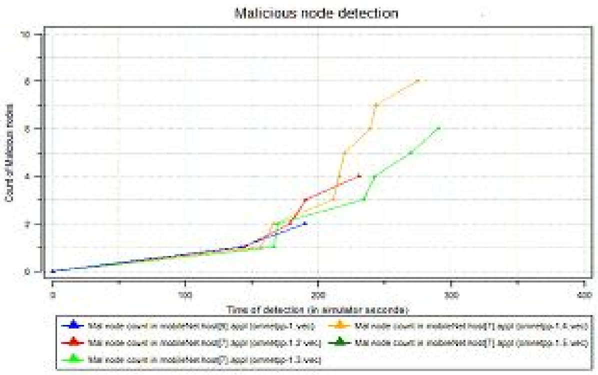

Figure 5. Intrusion Detection with varying malicious node count

With the settings used, no false positive identifications happened, even though the simulation ran for a considerable amount of time (in some cases more than double the time) after the actual malicious nodes were identified. Note that in the right-most run 90 percent of the nodes in the network are malicious. The graph shows 0 malicious nodes detected. This is perfectly expected behavior, since the Voting system proposed requires at least one neighbor node to vote. Here, none are available, as only one of the ten nodes is a valid node.

Conclusion

We aimed to determine a method to identify malicious or compromised nodes in a MANET with mobile nodes based on behavioral attributes. We proposed to use a system in which aberrations of normal behavior (or anomalies in behavior) are defined quantitatively by observing data exchange activity. We defined this anomaly in terms of these items:

- Long term behavior of a node as measured by continuous observation of its responses, mainly in the form of IDS logical layer acknowledgments to transmissions
- Short term response to transmissions we then selected NS-2 as the simulator of choice to create an environment dubbing real-life mobile nodes. Where there are mobile nodes, forwarding of data to the correct recipient cannot be done without the use of a routing algorithm. We used an implementation of AODV [17], an Ad hoc On-demand Distance Vector reactive routing protocol, to perform this function for us.

The last phase involved measurement of all the data with various simulation runs. A”Selective Forwarding Node” that simulates a”black hole” or”selective forwarding” attack by not replying to any transmissions is created. Such node also acts as”flooding attack” nodes. Instances of these nodes masquerade as malicious nodes.in the simulation runs. The data collected has shown that our proposed system works well. Our IDS can detect malicious nodes with almost 100 percent proficiency. The percentage of false positives is also reasonable, and does not exceed 20 percent for most simulation cases.

References

[1] Ns2 network simulator. http://www.isi.edu/nsnam/ns.

[2] A.Rajaram and S.Palaniswami. A trust based cross layer security protocol for ad hocnetworks. International Journal of Computer Science And Information Security, 6(1),2009

[3] Ashwin Perti ,Pradeep Sharma: Reliable AODV protocol for wireless Ad hoc Networking , 2009 IEEE International Advance Computing Conference, Patiala ,India(ICACC-2009,March 2009

[4] Amit Kumar Chandanan, Shailendra Kumar Shrivastava, “Secure Mobile Network Routing Protocol Using PSR,” cicn, pp.289-295, 2010 International Conference on Computational Intelligence and Communication Networks, 2010

[5] J. Binkley and W. Trost. Authenticated ad hoc routing at the link layer for mobile systems. *Wireless Networks*, 7(2):139–145, 2001

[6] Y. Hu, A. Perrig, and D. Johnson. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA’02)*, pages 3–13, June 2002.

[7] D. Johnson and D. Maltz. Dynamic source routing in ad-hoc wireless networks routing protocols. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[8] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *Proceedings of the ACM workshop on Wireless security (WiSE ’02)*, pages 21–30, September 2000

[9] F. Kargl, A. Klenk, S. Schlott, and M. Weber. Advanced detection of selfish or malicious nodes in ad hoc networks. In *Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, pages 152–165, August 200426

[10] Michael G Solomon and Mike Chappel. Information Security Illuminated. Jones and Bartlett, 2004

[11] Amitabh Mishra, Ketan Nadkarni, and Animesh Patcha. Intrusion detection in wireless ad hoc networks. IEEE wireless communications, February 2004

[12] Tao Li, Min Song, and Mansoor Alam. Compromized sensor node detection: A quantitative approach. IEEE International Conference on Distributed Computing Systems,pages 352–357, 2008.

[13] Chin-Yang Tseng, Poornima Balasubramanyam, Calvin K, Rattapon Limprasittiporn, and Karl Levitt Jeff Rowe. A specification-based intrusion detection system for aodv. pages 125–134, 2003.

[14] Ping Yi, Yichuan Jiang, Yiping Zhong, and Shiyong Zhang. Distributed intrusion detection for mobile ad hoc networks. Symposium on Applications and the Internet Workshop, 2005.

[15] Y.Zhang and W.Lee. Intrusion detection in wireless ad hoc networks. International Conference on Mobile Computing and Networks, page 275283, August 2000.

[16] Brent A. Peacock. Connecting the edge :mobile ad-hoc networks (manets) for network centric warfare. April 2007.

[17] C. Perkins and E. Royer. Adhoc On-demand Distance Vector Routing. IEEE workshop on Mobile Computing Systems and Applications, 3(4):90–100, February 1999

[18] Patrick Albers, Olivier Camp, Jean-Marc Percher1, Bernard Jouga, Ludovic Me, and Ricardo Puttini. Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches. 2005

[19] Joo B. D. Cabrera, Raman K. Mehra, and Carlos Gutirrez. Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad-hoc networks. International Conference on Mobile Computing and Networks, 9(1), January 2008

DOI: 01.IJNS.03.01.61